REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
12/30/2002

2. REPORT TYPE
Final Technical Report

3. DATES COVERED *(From – To)*
6/1/2001 - 9/30/2002

4. TITLE AND SUBTITLE
Interactive Display Models for Information Visualization in Virtual Reality

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-01-1-0917

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Maletic, Jonathan I.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Kent State University
P.O. Box 5190, Kent, Ohio, 44242-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING REPORT NUMBER

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research is investigating the use of Virtual Reality Environments (VE) to visualize complex data and information. Virtual Reality (VR) combines 3D views with animation to allow the user to navigate through the visualization. This application differs greatly from representing real worlds in VR for say training or demonstration. A generic display model for visualizing information in a VR is being developed. This model will extend previous research on 2D and 3D models. Information visualization is vital to the day-to-day operations of Navy personnel planning, distribution, and assignment.

15. SUBJECT TERMS
information visualization, software visualization, virtual reality

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON

19b. TELEPHONE NUMBER *(Include area code)*

103 174

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliance; e.g., 30-06-1998, xx-06-1998, xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written; e.g., Jun 1997 – Jun 1998, 1-10 Jun 1996, May – Nov 1998, Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report; e.g., F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report; e.g., AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report; e.g., 61101 A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report; e.g., 1F665702D1 257, ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report; e.g., 05, RF0330201, T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report; e.g., 001, AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas; e.g., Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization; e.g., BRL- 1234, AFWL-TR-85-4017-Vol-21-OT-2.

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR / MONITOR'S ACRONYM(S). Enter, if available; e.g., BRL, ARDEC, NADC.

11. SPONSOR / MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/monitoring agency, if available; e.g., BRL-TR-829, -215.

12. DISTRIBUTION / AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/restrictions or special markings are indicated, follow agency authorization procedures; e.g., RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classifications; e.g., U, C, S, etc. If this form contains classified information, stamp classification on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Final Technical Report

GRANT #: N00014-01-0917

PRINCIPLE INVESTIGATOR: Jonathan I. Maletic, Ph.D. (email: jmaletic@cs.kent.edu)

INSTITUTION: Kent State University (previously The University of Memphis)

GRANT TITLE: Interactive Display Models for Information Visualization in Virtual Reality

AWARD PERIOD: 6/1/2001 thru 9/30/2002

OBJECTIVE: The research is investigating the use of Virtual Reality Environments (VE) to visualize complex data and information. Virtual Reality (VR) combines 3D views with animation to allow the user to navigate through the visualization. This application differs greatly from representing real worlds in VR for say training or demonstration. A generic display model for visualizing information in a VR is being developed. This model will extend previous research on 2D and 3D models. Information visualization is vital to the day-to-day operations of Navy personnel planning, distribution, and assignment.

The goal of this research is the construction of a useful and generic information display model for use in advanced visualization environments. The display model will consist of a number of operators that act on data to transform it into visual representations in virtual reality. The operators will take advantage of human perception features not allowed by other mediums. These features include direction of motion, stereoscopic depth, and lighting directions. This research addresses the future directions of human computer interaction and interfaces, and information visualization.

APPROACH: An experimental software platform is being developed to test and evaluate the usability and suitability of the display model. The system will be developed to utilize advanced technology in virtual reality. A virtual reality development language will be utilized that is compatible with VRML and the CAVE virtual environment. The application here is to the visualization of large-scale software systems. A prototype software system is being developed as an experimental test bed for experimentation of the developed methods.

CONCLUSIONS: No visualization method addresses all the needs of the users. One successful approach to address more of the user's needs is to offer multiple views of the data as done by [Knight'01, Reiss'01, Storey'01]. Using one view of the data limits the number of attributes and the available exploration space. The solution we propose to overcome this problem is the efficient use of a 3D space for visualization.

Visualization in the 2D space has been actively explored. Many techniques for generating diagrams, graphs, and mapping information to the 2D representation have also been studied extensively. Although the question of what benefits 3D representation offer over 2D still remains to be answered, some experiments have given optimistic results. These results further motivate our work presented here.

The work of Hubona, Shirah and Fout [Hubona'97] suggest that users' understanding of a 3D structure improves when they can manipulate the structure. Ware and Franck [Ware'94] indicate that displaying data in three dimensions instead of two can make it easier for users to understand the data. In addition, the error rate in identifying routes in 3D graphs is much smaller than 2D [Ware'93]. The CyberNet system [Dos Santos'00] shows that mapping large amount of (dynamic) information to 3D representation is beneficial, regardless of the type of metaphors (real or virtual) used. Also, 3D representations have been shown to better support spatial memory tasks than 2D [Tavanti'01]. In addition, the use of 3D representations of software in new mediums, such as virtual reality environments, are starting to be explored [Knight'99, Maletic'01].

MacKinlay [MacKinlay'86] defined two criteria to evaluate the mapping of data to a visual metaphor: expressiveness and effectiveness. These criteria were used in 2D mappings, but can also be applied for 3D mappings. *Expressiveness* refers to the capability of the metaphor to visually represent all the information we desire to visualize. For instance, if the number of visual parameters available in the metaphor for displaying information is fewer than the number of data values we wish to visualize, the metaphor will not be able to meet the expressiveness criterion. The relationship between data values and visual parameters has to be a univocal relationship; otherwise, if more than one data value is mapped onto the same visual parameter than it will be impossible to distinguish one value's influence from the other. On the other hand, there can always be visual parameters that are not used to map information, as long as there is no need for them to be utilized.

The second criterion, *effectiveness*, relates to the efficacy of the metaphor as a means of representing the information. Along the effectiveness dimension we can further distinguish several criteria: effectiveness regarding the information passing as visually perceived, regarding aesthetic concerns, regarding optimization (e.g., number of polygons needed to render the view). In the case of quantitative data (e.g., software metrics, LOC, trace data), not only the number of visual parameters has to be sufficient to map all the data, but also, they must be able to map the right data. There are visual parameters that are not able to map a specific category of data; for instance, shape is not useful for mapping quantitative data, while the size of a metaphor is adequate. Effectiveness implies the categorization of the visual parameters according to its capabilities of encoding the different types of information. Moreover, this also implies categorizing the information according to its importance so that information that is more important can be encoded more efficiently when options must be taken. This categorization of the importance of the information has two expressions: one is an assigned importance of the information in the context of a software system; the other is a preference of the user. Nonetheless, the user may choose to override this and define his own importance of the data, according to his priorities is usually the first step to understand a phenomenon or system. Although these characteristics of data apply mostly to data visualization, they must be taken into consideration when visualizing a software system.

The prototype we developed, IMSOvision, automatically translates given source code and generates the VRML source for the visualization. It uses a visual display language to incorporate a flexible abstraction mechanism. This will allow for better navigation of the very large graphs that represent software system. We are utilizing an abstraction mechanism similar to what Colin Ware used to represent large-scale graphs. Modules or subsystems are represented encased in semi-transparent cubes. These can be shrunk and enlarged with respect to the current cognitive importance. The contents of the cube will always be visible but in the minimized mode only a thumb nail of the internals is discernable. Additionally, we have added the names of classes and attributes to the visualization. This system is working, however it is a prototype and much work is necessary to have it fully functional.

We have encountered two interesting sub-problems that require further research. The first is the development of a 3D layout algorithm that readily support large complex graphs. A variety of graph layout algorithms exist, but few that apply to a 3D space. Additionally, the domain (in this case software) may allow the nodes to be weighted by overall importance. This weighting could vary based on the particular task. The second problem is defining a natural abstraction mechanism to support better navigation of large graphs.

SIGNIFICANCE: The significance of this research is two fold. Firstly, it is investigating how to visualize large complex graphs in immersive virtual reality. This new visual medium should prove to be very powerful for this general application. Secondly, the research is addressing the complex domain of software visualization. Any tool or technique that provides benefits to software developers has the potential for large cost savings and quality improvement. Practical software visualization must provide tools to select and display just the information of interest. A practical software visualization system can be achieved by focusing on abstractions. We view the problem of software visualization within these five dimensions: 1) Target – what to represent? 2) Medium – where to represent? 3) Representation – how to represent? 4) Audience – who will use the visualization? 5) Tasks – why is the visualization needed?

There is variety of information with respect to software that can be represented. Metric information, static call graphs, runtime profiles, and trace information are all possible targets for visual representation. The type of information needed is in direct relation to the task. The type of medium we display the information on can greatly vary. In our work, we are investigating using mediums besides the 21-inch monitor. The need for more screen restate is motivating much research into new display technology. In addition, the use of multi-type mediums (e.g., laptop in a VE) could be of benefit. In general, a software visualization system should determine the abstraction level of the information it depicts about the software system. It should use a visual language or mapping to translate source code (and possibly external documentation) into a visual representation. The choice of mapping depends on the type of information it represents and the media used in the representation. The user tasks (i.e., manipulation, navigation, etc.) that the system supports, including program comprehension tasks, should be specified. In addition, the representation must support multi-views and multi-levels of abstractions. Lastly, there are very different types of audiences and a wide range of software engineering tasks that a software visualization environment can be directed towards. Audiences may range from teaching novice programmers better development techniques to expert developers maintaining a 20 million-line program. This dimension is directly tied with the types of tasks, from debugging and optimization to re-engineering and maintenance.

The DoD must maintain and manage a very large amount of software. Tools the support these tasks are vital for understanding the quality and capabilities of these software systems. This research is directly addressing these issues. Additionally, this type of visualization framework may be applied to other domains for example, visualizing the relationships and interaction of (non-software) systems or products used by the DoD. We believe this could be used to visualize the interacting components of complex systems such as a ship.

PATENT INFORMATION: No new patents or patent applications were generated as a result of this grant.

AWARD INFORMATION: No major awards, however the PI did leave the University of Memphis to accept a new position at Kent State University.

PERSONNEL SUPPORTED BY GRANT

The following graduate students were supported in part by this grant.

1. Gregory T. Dunlap, The University of Memphis, completed M.S. in Computer Science 12/2001
2. Andrian Marcus, Kent State University, will complete Ph.D. in Computer Science in 2003
3. Michael Collard, Kent State University, will complete Ph.D. in Computer Science in 2004
4. Louis Feng, Kent State University, will compute M.S. in Computer Science in 2003

REFEREED PUBLICATION

1. Maletic, J., Marcus, A., Collard, M.L., "A Task Oriented View of Software Visualization Taxonomies for Software Visualization Revisited", in Proceedings of the IEEE Workshop on Visualizing Software for Understanding and Analysis (VISSOFT'02), Paris, France, June 26th, 2002, pp. 32-40.

2. Marcus, A., Maletic, J.I., "Identification of High-Level Concept Clones in Source Code", in Proceedings of the 16th IEEE International Conference on Automated Software Engineering (ASE 2001), San Diego, CA, USA, November 26-29, pp. 107-114.

3. Marcus, A., Maletic, J.I., Lin, K.I. "Ordinal Association Rules for Error Identification in Data Sets", in Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM 2001), Atlanta, GA, USA, November 5-10, pp. 589-591.

4. Collard, M.L., Maletic, J.I., Marcus, A., "Supporting Document and Data Views of Source Code", in Proceedings of the 2nd ACM Symposium on Document Engineering (DocEng'02), McLean, VA, November 8-9, pp. 34-41.

5. Maletic, J.I., Collard, M.L., Marcus, A., "Source Code Files as Structured Documents", in Proceedings of the 10th IEEE Workshop on Program Comprehension (IWPC 2002), Paris France, June 27-29, 2002, pp. 289-292.

6. Maletic, J., Leigh, J., Dunlap, G., Marcus, A., "A System for Visualizing Object Oriented Software Systems in Virtual Reality", in Proceedings of the 9th IEEE International Workshop on Program Comprehension (IWPC'01), Toronto, Ontario, May 12-13th, 2001, pp. 26-35.

7. Maletic, J., Leigh, J., Marcus, A., "Visualizing Software in an Immersive Virtual Reality Environment", in the Proceedings for the ICSE Workshop on Software Visualization, Toronto, Ontario, May 13-14th, 2001, pp. 49-54.

BOOK CHAPTERS, SUBMISSIONS, ABSTRACTS, AND OTHER PUBLICATIONS

Thesis:

1. Gregory Thomas Dunlap, *Visualizing Object Oriented Software in a Virtual Reality Environment*, M.S. Thesis, Division of Computer Science, The University of Memphis, Memphis TN, December 2001.

2. Feng, Louis, *Source Viewer 3D (sv3D): A Framework for Software Visualization*, M.S. Thesis, Department of Computer Science, Kent State University, Kent, Ohio, (*to be completed May 2003*).

Submissions (under review):

1. Marcus, A., Feng, L., Maletic, J.I., "3D Representations for Software Visualization" submitted to SoftVis 2003.

2. Marcus, A., Feng, L., Maletic, J.I., "Support for Program Comprehension using 3D Visualizations" submitted to IWPC 2003.